



TITLE:

有理関数近似の離散化における問題点 (数学解析の計算機上での理論的展開とその遂行可能性)

AUTHOR(S):

村上, 裕美; 甲斐, 博; 野田, 松太郎

CITATION:

村上, 裕美 ...[et al]. 有理関数近似の離散化における問題点 (数学解析の計算機上での理論的展開とその遂行可能性). 数理解析研究所講究録 2002, 1286: 34-50

ISSUE DATE:

2002-09

URL:

<http://hdl.handle.net/2433/42459>

RIGHT:

有理関数近似の離散化における問題点

愛媛大学 理工学研究科 村上 裕美 (Yumi MURAKAMI) *

愛媛大学 工学部 甲斐 博 (Hiroshi KAI)[†]

愛媛大学 工学部 野田 松太郎 (Matu-Tarow NODA)[‡]

1 はじめに

有理関数補間を用いて関数近似を行った場合、補間区間内に不必要な極が現れる場合があるという問題がある [3]。この不必要な極は、補間を行う有理関数の分子分母の多項式が、補間区間内に非常に近い値の零点を持つために現れる。これまでの研究では、このことを利用して分子分母の多項式の近似 GCD を求めて、この零点を近似的な共通因子として取り除く方法 [1, 2] が提案されている。本研究では、有理関数補間を行うときに現れる不必要な極を生じる原因の解明を目的として、素朴な有理関数近似の計算に関する再検討を行うとともに、安定化理論 [4] を用いた有理関数補間についての検討を行った。

2 素朴な有理関数近似

関数 $f(x) \in C[a, b]$ に対する有理関数補間は次のように計算される。有限個の離散点 $a = x_0 < x_1 < \cdots < x_{m+n} = b$ を与え、対応する関数値 $f(x_k) = f_k$, $k = 0, 1, \dots, m+n$ を求める。ここで与えられた m, n に対して、

$$r(x_k) = \frac{p(x_k)}{q(x_k)} = f_k \quad k = 0, 1, \dots, m+n$$

を満たすような

$$r_{m,n}(x) = \frac{p_m(x)}{q_n(x)} = \frac{\sum_{i=0}^m a_i x^i}{\sum_{j=0}^n b_j x^j}$$

を求める。この有理関数を (m, n) 有理関数と呼び、便宜上 $b_0 = 1$ と規格化する。多項式の係数 a_i, b_j は一般に浮動小数であり、以下のような連立一次方程式を解くことによって求められる。

*cumi@hpc.cs.ehime-u.ac.jp

†kai@cs.ehime-u.ac.jp

‡noda@cs.ehime-u.ac.jp

$$\begin{pmatrix} 1 & x_0 & \cdots & x_0^m & -f_0x_0 & \cdots & -f_0x_0^n \\ 1 & x_1 & \cdots & x_1^m & -f_1x_1 & \cdots & -f_1x_1^n \\ 1 & x_2 & \cdots & x_2^m & -f_2x_2 & \cdots & -f_2x_2^n \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 1 & x_m & \cdots & x_m^m & -f_mx_m & \cdots & -f_mx_m^n \\ 1 & x_{m+1} & \cdots & x_{m+1}^m & -f_{m+1}x_{m+1} & \cdots & -f_{m+1}x_{m+1}^n \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 1 & x_{m+n} & \cdots & x_{m+n}^m & -f_{m+n}x_{m+n} & \cdots & -f_{m+n}x_{m+n}^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_m \\ b_1 \\ \vdots \\ b_n \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_m \\ f_{m+1} \\ \vdots \\ f_{m+n} \end{pmatrix}$$

2.1 素朴な有理関数近似の問題点

前節で述べたような有理関数を用いて関数近似を行うと、元の関数 $f(x)$ が連続であるのに対して得られた有理関数が不必要な極を持ち、不連続になってしまう場合がある。これは、不必要な極に対応する有理関数の分子 $p_m(x)$ の零点が分母 $q_n(x)$ の零点に非常に近い値をとっていることが原因となっている [2]。例えば、関数 $\log(x+2)$ を補間区間を $[-1, 1]$ の間で分子分母の次数が 4 次 ($m=n=4$) の有理関数で近似することを考える。有効桁数 7 桁で連立一次方程式を解くと、次のような有理関数が得られる。

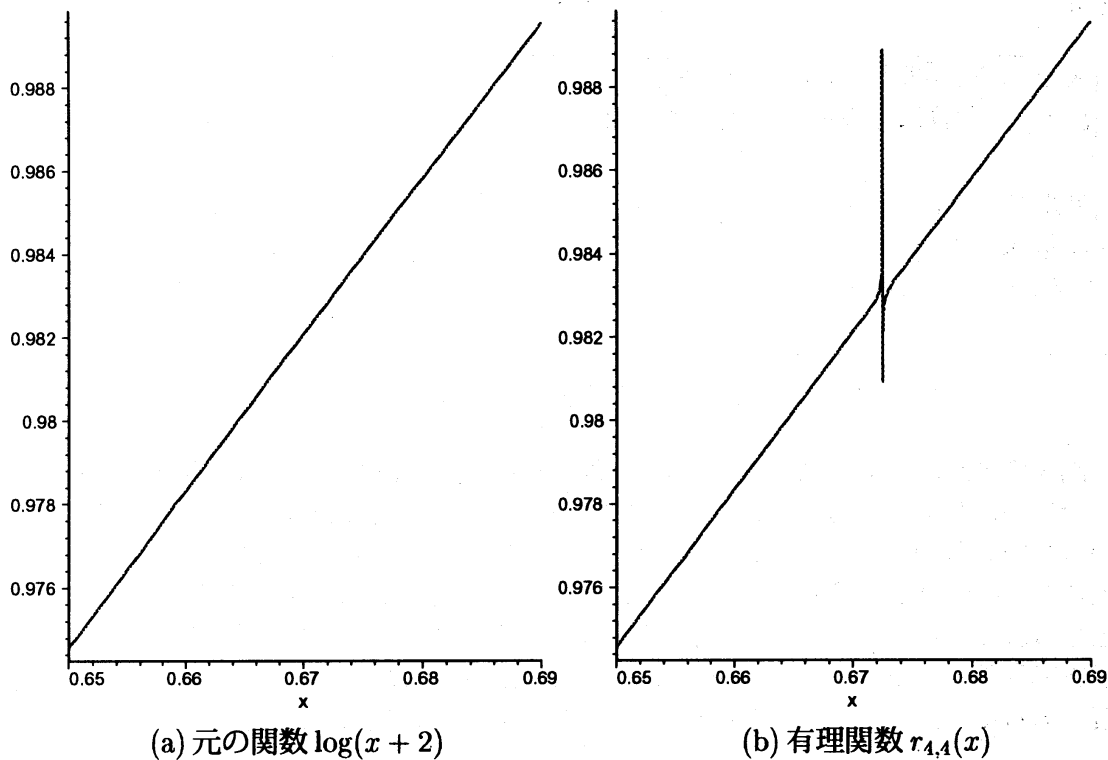


図 1: 有効桁数 4 桁で得られた係数による有理関数近似

$$\begin{aligned}
r_{4,4}(x) &\simeq \frac{0.6931471 + 0.01180017x - 1.168640x^2 - 0.5353216x^3 - 0.04802865x^4}{1 - 0.7043236x - 0.9975914x^2 - 0.2398265x^3 - 0.01131871x^4} \\
&= \frac{-0.04802865(x + 8.201649)(x + 2.616697)(x + 0.9999999)(x - 0.6724661)}{-0.01131871(x + 15.91957)(x + 3.727162)(x + 2.214231)(x - 0.6724660)}
\end{aligned}$$

この有理関数では、補間区間内に存在する分母の零点 0.6724660 と非常に近いところに分子の零点 0.6724661 が存在している。このような有理関数を用いて関数近似を行うと、この近似的に近い値の零点の付近に不必要な極が現れる (図 1)。

3 ハイブリッド有理関数近似

有理関数近似を行った際に現れる不必要な極を除去し、高精度な近似を得る方法の一つにハイブリッド有理関数近似がある [1, 2, 3]。ハイブリッド有理関数近似は、有理関数の分子分母に存在する近似的に近い値の零点を、近似 GCD を用いて共通因子として取り除くことで、分子分母が近似的な共通因子を持たないような有理関数を構成する方法である。ハイブリッド有理関数近似のアルゴリズムを図 2 に示す。図 2 のアルゴリズムの中で用いられる近似 GCD を求めるには図 3 や図 4 のアルゴリズムを利用する。

[入力] : 有理関数 $r_{m,n}(x) = \frac{p_m(x)}{q_n(x)}$
[出力] : 共通因子を除去した有理関数 $\tilde{r}(x) = \frac{\tilde{p}(x)}{\tilde{q}(x)}$

[アルゴリズム] :

1. $AppGCD(p_m(x), q_n(x)) = g(x)$
 2. $\tilde{r}(x) = \frac{p_m(x)/g(x)}{q_n(x)/g(x)}$
-

図 2: ハイブリッド有理関数近似のアルゴリズム

図 3 のアルゴリズムは、Euclid の互除法を浮動小数点係数に対応できるように拡張したアルゴリズムとなっている。図 4 のアルゴリズムは、入力の多項式の根が既に分かっているものとしてその根の値を比較し、ある一定の精度 δ によって近いと判断されたものについてはその根の値 (y_{ik}, z_{jk}) の中点を取り、この値を近似 GCD $\tilde{d}_\delta(x)$ の根となるようにして近似 GCD を構成するようなアルゴリズムになっている。ハイブリッド有理関数近似では、分子分母の多項式の近似的な共通因子だけを取り除くことが必要であるため、入力の多項式の根の値を直接比較して近似 GCD を求める図 4 のアルゴリズムの方が適している。

[入力] : 多項式 $P_1(x), P_2(x)$
 [出力] : 近似 GCD $AppGCD(P_1(x), P_2(x)) = g(x)$

[アルゴリズム] :

1. $F \leftarrow P_1(x), G \leftarrow P_2(x)$
 2. $F = QG + \max(1, \|Q\|)R$ を満たす Q, R を求める。
 ($\|Q\|$: 多項式 Q の係数の絶対値の最大値)
 3. if all coefficients of $R \leq \epsilon$
 then $AppGCD(P_1(x), P_2(x)) = R$
 else $F \leftarrow G, G \leftarrow R$ go to step2.
-

図 3: 近似 GCD のアルゴリズム

[入力] : 多項式 $P_1(x) = u \times \prod_{i=1}^m (x - y_i), P_2(x) = v \times \prod_{j=1}^n (x - z_j)$, 精度 δ
 [出力] : 近似 GCD $\delta - gcd : \tilde{d}_\delta(x)$

[アルゴリズム] :

- $\tilde{d}_\delta(x) = \prod_{k=1}^r (x - x_k), \quad x_k = \frac{y_{ik} + z_{jk}}{2} \quad (k = 1, 2, \dots, r)$
 y_{ik}, z_{jk} はそれぞれ多項式 $P_1(x), P_2(x)$ の根で、 $|y_{ik} - z_{jk}| \leq 2\delta$ を満たすもの。
-

図 4: Pan の近似 GCD のアルゴリズム

3.1 ハイブリッド有理関数近似の例

2.1 節の例で使した有理関数 $r_{4,4}(x)$ を用いてハイブリッド有理関数近似を行った例を示す。図 3 のアルゴリズムで近似 GCD の値を求めると、

$$AppGCD(p_4(x), q_4(x)) = g(x) \simeq -0.008251166x + 0.005548411$$

となる。この近似 GCD の値で元の関数を割ると、以下のような $\tilde{r}(x)$ が得られる。

$$\begin{aligned} \tilde{r}(x) &= \frac{p_4(x)/g(x)}{q_4(x)/g(x)} \\ &\simeq \frac{5.820832x^3 + 68.79246x^2 + 187.8921x + 124.9160}{1.371771x^3 + 29.98820x^2 + 141.0683x + 180.2204} \\ &= \frac{5.820832(x + 8.201647)(x + 2.616740)(x + 0.9999340)}{1.371771(x + 15.91957)(x + 3.727228)(x + 2.214140)} \end{aligned}$$

この \tilde{r} を用いて関数近似を行うと、不必要な極のない高精度な近似を得ることができる (図 5)。

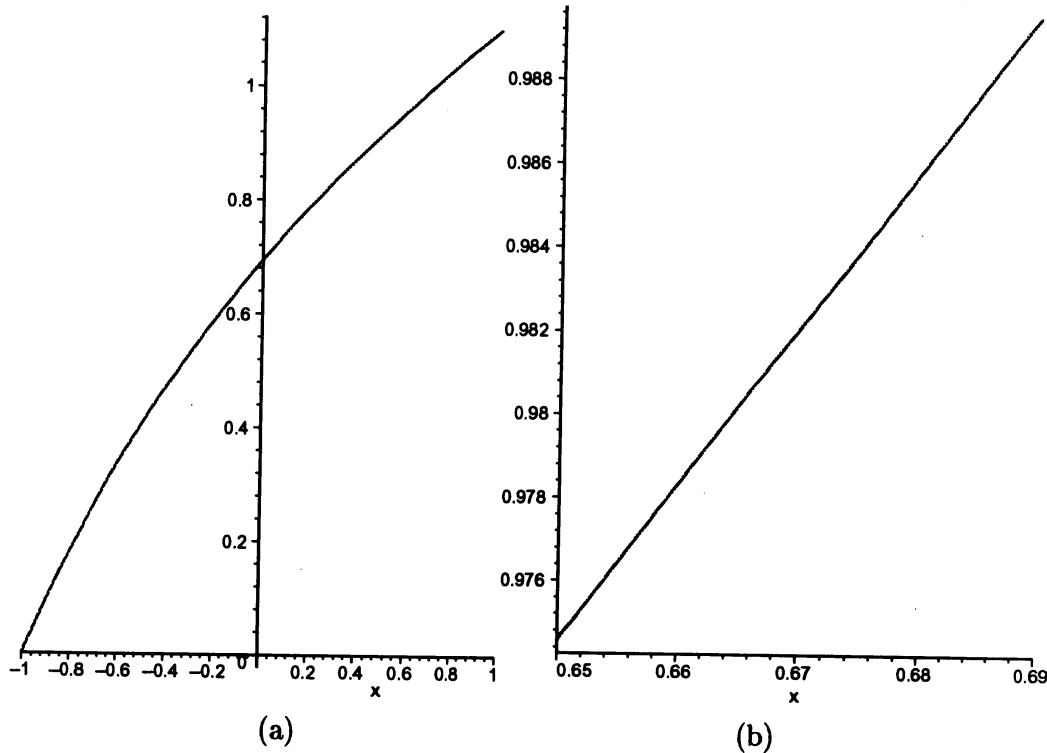


図 5: ハイブリッド有理関数近似

図 5 の (a) は、補間区間 $[-1, 1]$ 全体を近似したもので、(b) は図 1 と同じ範囲を拡大したものである。この図から、元の関数に対して不必要な極のない高精度な近似が得られることが確認できる。

4 有理関数近似の例

4.1 不必要な極のふるまい

ここでは、有効桁数は補間多項式の次数を変化させたときの不必要な極のふるまいについて再検討を行う。 $\log(x + 2)$ を補間区間 $[-1, 1]$ で有理関数近似を行った場合の不必要な極の位置を表 1 に示す。表で $r(5, 5)$ と書かれているものは、分子分母の次数が 5 次の有理関数、 $r(10, 10)$ ならば分子分母が 10 次の有理関数で近似することを意味する。この結果から、極の位置に関しては規則性がないが、有効桁数が少ない場合や補間を行う多項式の次数が大きい場合に不必要な極が出やすくなっていることがわかる。表 1 では補間点を等間隔にとった場合の不必要な極の位置を示しているが、補間点の取り方を変えた場合でも、不必要な極は位置が異なるものの、同じような特徴が見られる。

表 1: 不必要な極の位置 ($\log(x+2)$, 補間区間 $[-1, 1]$)

有効桁	$r(5, 5)$	$r(10, 10)$	$r(15, 15)$	$r(20, 20)$	$r(25, 25)$
10 桁	-0.080667	-0.66028 -0.87680	-0.54814	-0.17765 0.59266	0.74821
20 桁	-	0.35248	-0.37456 0.54999 0.72561	-0.098106 0.92383	0.24963
30 桁	-	-	-0.56625	-	0.29373
40 桁	-	-	-	0.81169	-
50 桁	-	-	-	-0.78729	0.96187
60 桁	-	-	-	-	-0.75259

4.2 行列の条件数

前節の結果から、同じ次数の有理関数で関数近似を行った場合でも、有効桁数によって不必要な極の位置が変化していることがわかる。また、有効桁数を上げていくと不必要な極は現れなくなる。そこで、Gauss 消去法を適用する係数行列の条件数を求め、行列の性質と不必要な極の関係についての検証を行った。表 1 で使用したのと同じ関数 $\log(x+2)$ で、同じ補間区間 $[-1, 1]$ とした場合の行列の条件数を、図 6 に示す。条件数は $\text{cond } A = \|A\| \|A^{-1}\|$ によって求めている。

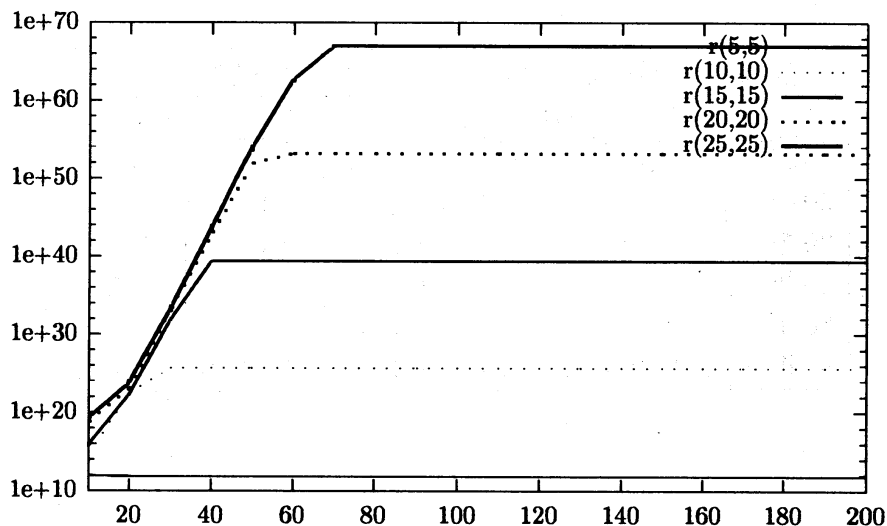
図 6: $\log(x+2)$ を有理関数近似する場合の係数行列の条件数

図 6 からわかるように、有理関数補間に使用される行列は非常に悪条件となってい

る。また、行列の条件数は有効桁数の少ない場合には有効桁ごとに条件数の値が全く異なっているが、有効桁数を多くするにしたがって一定の値に収束してくることがわかる。条件数の変化からも明らかだが、分子分母に5次のような低次の多項式を持つような有理関数補間においても、有効桁数が少ない場合には各係数の値は不安定となり、当然得られる有理関数も不安定になる。一方、このような場合にも有効桁数を20桁以上にすると、各係数の値も得られる有理関数の値も安定してくることがわかる。このような安定な有理関数補間が得られるのは、図6から $r(10, 10)$ 有理関数では有効桁数40桁、 $r(15, 15)$ 有理関数では60桁等となることがわかる。ここで、条件数は、残差が何倍拡大されて相対誤差に反映するかの倍率となっていることに注目する。分子分母が5次の有理関数では、条件数が 10^{11} 程度で安定していることから、10進数にして10桁で計算を行うと、得られた解の精度はほとんど1桁もないといえる。図1で、有効桁10桁では不必要な極が生じ、20桁以上では不必要な極が現れないのは、計算に使用する有効桁数が条件数 10^{11} に対して十分な精度を保つことができる大きさになっているためであると考えられる。

ここで、少ない有効桁数では得られる解の値が不安定であっても、結果として得られた有理関数で近似を行うと図1に示されるように不必要な極を生じるが、それ以外の部分は正しく関数近似を行うことができる。したがって、ハイブリッド有理関数近似を利用して不必要な極を除去すれば、高精度な近似が得られる。ハイブリッド有理関数近似により不必要な極を除去した場合の元の関数との誤差を表2に示す。誤差は、補間区間 $[-1, 1]$ を100分割した点 x_k , ($k = 1, 2, \dots, 100$)を用いて、次のような方法で求めた。

$$E = \max(|\log(x_k + 2) - \tilde{r}(x_k)|), (k = 1, \dots, 100)$$

表 2: 元の関数との誤差 ($\log(x + 2)$, 補間区間 $[-1, 1]$)

有効桁	$r(5, 5)$	$r(10, 10)$	$r(15, 15)$	$r(20, 20)$	$r(25, 25)$
10 桁	9.5e-9	1.0e-9	3.0e-8	2.6e-9	4.0e-8
20 桁	1.9e-10	2.7e-16	1.6e-17	3.0e-18	3.6e-18
30 桁	1.9e-10	1.5e-19	6.3e-24	1.6e-26	6.5e-26
40 桁	1.9e-10	1.5e-19	1.3e-28	3.1e-31	4.7e-33
50 桁	1.9e-10	1.5e-19	1.3e-28	1.4e-35	2.2e-43
60 桁	1.9e-10	1.5e-19	1.3e-28	9.8e-38	6.5e-47

4.3 残差

前節では、有効桁数が少ないときには得られる解が不安定であるが、不必要な極を除けば正しい有理関数近似が得られることを前節で示した。そこで、有効桁数が少ない場合に得られる解の正当性を検証するために、残差の値を求めた。

表3の結果から、解の値が不安定となっている有効桁数の少ない部分でも、不必要な極を除けば残差の値を見ると十分な精度で正しく解けていると判断することができる。なお、解が不安定となる原因としては、入力時の浮動小数近似による丸め誤差の影響によって、Gauss消去法の計算過程で行われるピボット選択の選択場所が異なっていることなどが考えられる。

表 3: 残差 $\|Ax - b\|$ ($\log(x + 2)$, 補間区間 $[1, 2]$)

有効桁	$r(5, 5)$	$r(10, 10)$	$r(15, 15)$	$r(20, 20)$	$r(25, 25)$
10 桁	1.4e-8	2.3e-9	6.2e-8	1.3e-8	1.5e-8
20 桁	4.2e-19	2.3e-18	6.5e-19	2.3e-18	1.7e-18
30 桁	1.8e-29	4.2e-28	2.0e-28	1.0e-28	3.4e-28
40 桁	2.5e-39	6.0e-39	5.9e-38	7.3e-39	1.4e-38
50 桁	2.4e-49	7.8e-49	2.5e-48	1.0e-47	2.0e-48
60 桁	1.1e-59	4.8e-59	6.6e-58	9.7e-58	1.6e-57

4.4 対称関数の有理関数近似

補間区間内で対称となるような関数の近似を行う場合、補間を行う有理関数の分子分母の次数によっては、係数行列の行列式が厳密には0になるような場合がある。このような行列に対して厳密計算を行うと、Gauss消去法の計算過程でランク落ちが生じるため、解を一意に求めることはできない。ところが、浮動小数近似した値を利用すると、浮動小数近似による誤差からランク落ちを生じず、解が一意に求まってしまう場合がある。このような関数の例としては、Rungeの例として知られる $1/(25x^2 + 1)$ のような関数や $\cos(x)$ などの関数が挙げられる。厳密には行列式の値が0となりランク落ちを生じるこのような関数では、浮動小数近似することで行列式の値がわずかに0から離れた値となるので、条件数の値は有効桁数を上げるほど急激に悪化することになる。

このような浮動小数近似された値を利用して Gauss 消去法を行うと、本来はランク落ちを引き起こすはずの成分に誤差が含まれることで、非常に小さな値となり、連立一次方程式の解が一意に求まる。こうして求まった有理関数は、どれも不必要な極を生じる部分以外は正しく元の関数の近似できるものになっている。このような不必要な極を持つ関数は、ハイブリッド有理関数近似を用いて分子分母の共通因子を除去すれば、不必要な極のない高精度な有理関数を構成することができるので、本来は厳密計算では求めることができないような有理関数を、浮動小数近似することによって求めることができるようになる。この過程を、Rungeの例として知られる関数 $1/(25x^2 + 1)$ を例として以下に解析する。

4.4.1 Runge の例の場合

Runge の例は、元の関数が有理関数の形をしているため、得られる関数が元の有理関数と一致することが理想的である。ここで、分子 m 次、分母 n 次の有理関数を $r_{m,n}(x)$ と表すとする、 $r_{1,2}(x), r_{2,2}(x)$ による近似は、実際に元の関数と完全に一致する。次に、元の有理関数よりも明らかに次数の大きな有理関数 $r_{3,3}(x)$ や $r_{4,4}(x)$ などを用いて近似を行うことを考える。

1. 厳密計算を行った場合

厳密計算で $r_{3,3}(x)$ による近似を求めようとする、Gauss 消去法の計算過程で次のようなランク落ちが生じる。

$$\left[\begin{array}{cccc|cccc} 1 & -1 & 1 & -1 & \frac{1}{26} & -\frac{1}{26} & \frac{1}{26} & \frac{1}{26} \\ 0 & 2 & 0 & 2 & -\frac{1}{13} & 0 & -\frac{1}{13} & 0 \\ 0 & 0 & -1 & 0 & 0 & \frac{1}{26} & 0 & \frac{25}{26} \\ 0 & 0 & 0 & -\frac{10}{27} & -\frac{125}{4251} & -\frac{250}{12753} & \frac{5}{4251} & -\frac{6250}{12753} \\ 0 & 0 & 0 & 0 & -\frac{1250}{24089} & -\frac{2050}{216801} & \frac{50}{24089} & -\frac{51250}{216801} \\ 0 & 0 & 0 & 0 & 0 & -\frac{100}{1989} & 0 & -\frac{2500}{1989} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right]$$

ここで、未定係数 a を利用すれば、厳密計算で得られる $r_{3,3}(x)$ は次のようになる。

$$r_{3,3}(x) = \frac{25 + ax}{(1 + 25x^2)(25 + ax)}$$

未定係数の含まれている項 $(25 + ax)$ は分子分母の共通因子となっており、簡約化すれば Runge の例の元の関数に一致することは明らかである。

次に、ランク落ちを生じている最後の行の対角成分と対応する右辺の値を記号 a, b と置き換え、以下のような行列を構成する。

$$\left[\begin{array}{cccc|cccc} 1 & -1 & 1 & -1 & \frac{1}{26} & -\frac{1}{26} & \frac{1}{26} & \frac{1}{26} \\ 0 & 2 & 0 & 2 & -\frac{1}{13} & 0 & -\frac{1}{13} & 0 \\ 0 & 0 & -1 & 0 & 0 & \frac{1}{26} & 0 & \frac{25}{26} \\ 0 & 0 & 0 & -\frac{10}{27} & -\frac{125}{4251} & -\frac{250}{12753} & \frac{5}{4251} & -\frac{6250}{12753} \\ 0 & 0 & 0 & 0 & -\frac{1250}{24089} & -\frac{2050}{216801} & \frac{50}{24089} & -\frac{51250}{216801} \\ 0 & 0 & 0 & 0 & 0 & -\frac{100}{1989} & 0 & -\frac{2500}{1989} \\ 0 & 0 & 0 & 0 & 0 & 0 & a & b \end{array} \right]$$

これに対して後退代入を行い、有理関数の係数を求めると次のような有理関数が得られる。

$$\begin{aligned} r_{3,3}(x) &= \frac{1 + \frac{b}{25a}x}{1 + \frac{b}{25a}x + 25x^2 + \frac{b}{a}x^3} \\ &= \frac{25a + bx}{(1 + 25x^2)(25a + bx)} \end{aligned}$$

記号 a, b が含まれる項は、分子分母の共通因子となっており、簡約化すれば元の Runge の例の関数と一致することが分かる。

2. 数値計算を行った場合

数値計算によって $r_{3,3}(x)$ を求めようとする、浮動小数近似による誤差から、本来はランク落ちを起こすはずの成分に微小な値が残り、一意に解を求めることができる。有効桁 5 桁で計算した場合は、次のようになる。

$$\left[\begin{array}{cccccccc|c} 1.0 & -1.0 & 1.0 & -1.0 & 0.038462 & -0.038462 & 0.038462 & 0.038462 \\ 0 & 2.0 & 0 & 2.0 & -0.076924 & 0 & -0.076924 & 0 \\ 0 & 0 & -1.0 & 0 & 0 & 0.038462 & 0 & 0.96154 \\ 0 & 0 & 0 & -0.3704 & -0.029402 & -0.019602 & 0.0011756 & -0.49004 \\ 0 & 0 & 0 & 0 & -0.051899 & -0.0094565 & 0.0020770 & -0.23643 \\ 0 & 0 & 0 & 0 & 0 & -0.050274 & 0.179 \text{ e-6} & -1.2569 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.880 \text{ e-5} & 0.539 \text{ e-4} \end{array} \right]$$

ランク落ちを起こすはずの最後の行に、有効桁 5 桁に対して微小な値 -0.88008 e-6 、 0.53991 e-4 が含まれている。これによって、有理関数の係数は一意に求まり、次のようになる。

$$\begin{aligned} r_{3,3}(x) &\simeq \frac{1.0000 - 2.4541x + 0.5 \text{ e-4 } x^2 + 0.9 \text{ e-4 } x^3}{1 - 2.4550x + 25.001x^2 - 61.348x^3} \\ &= 0.1467 \text{ e-5} \frac{(x - 0.40749)(x + 165.61)(x - 164.65)}{(x - 0.40749)(x^2 - 0.3798 \text{ e-4 } x + 0.040004)} \end{aligned}$$

$r_{3,3}(x)$ の分子分母には、 $(x - 0.40749)$ という近似的な共通因子が存在していることがわかる。これが、不必要な極を生じる原因となる近似的な共通因子である。近似 GCD を用いてこれを除去すれば、

$$\tilde{r}(x) \simeq \frac{-2.4541}{-61.348x^2 - 2.4550} = \frac{1.0}{24.998x^2 + 1.0004}$$

となり、元の Runge の例の関数 $1/(25x^2 + 1)$ に対して高精度な近似になっていることが分かる。

次に、厳密計算の場合と同様にランク落ちを引き起こすはずの成分に a, b という記号を代入することを考える。ここでは、 -0.88008 e-6 , 0.53991 e-4 をそれぞれ a, b で置き換えることになる。

$$\left[\begin{array}{cccccccc|c} 1.0 & -1.0 & 1.0 & -1.0 & 0.038462 & -0.038462 & 0.038462 & 0.038462 \\ 0 & 2.0 & 0 & 2.0 & -0.076924 & 0 & -0.076924 & 0 \\ 0 & 0 & -1.0 & 0 & 0 & 0.038462 & 0 & 0.96154 \\ 0 & 0 & 0 & -0.3704 & -0.029402 & -0.019602 & 0.0011756 & -0.49004 \\ 0 & 0 & 0 & 0 & -0.051899 & -0.0094565 & 0.0020770 & -0.23643 \\ 0 & 0 & 0 & 0 & 0 & -0.050274 & 0.179 \text{ e-6} & -1.2569 \\ 0 & 0 & 0 & 0 & 0 & 0 & a & b \end{array} \right]$$

これに対して後退代入を行うと、以下のような有理関数が得られる。

$$\frac{a + 0.107 \text{ e-}3 ax + 0.04bx + 0.5 \text{ e-}4 ax^2 + 0.137 \text{ e-}6 bx^2 - 0.1 \text{ e-}3 ax^3 - 0.3 \text{ e-}5 bx^3}{a + 0.0002ax + 0.040bx + 25.001ax^2 + 0.35772 \text{ e-}5 bx^2 + bx^3}$$

このままでは分子分母に共通因子は存在しない。ここで、有効桁 5 桁であることを考慮して微小な係数を無視すると、

$$\tilde{r}_{3,3}(x) \simeq \frac{a + 0.04bx}{a + 0.04bx + 25ax^2 + bx^3} = \frac{(25.0a + 1.0bx)}{(1.0 + 25.0x^2)(25.0a + 1.0bx)}$$

となり、厳密計算を行った場合と同じ結果が得られる。また、ここで適当な値、例えば $a = 0.01, b = 50$ を代入すると、

$$\begin{aligned} r_{3,3}(x) &\simeq \frac{1. + 200.02x + 0.73795 \text{ e-}3 x^2 - 0.015100x^3}{1 + 200.10x + 25.019x^2 + 5000.0x^3} \\ &= -0.302 \text{ e-}5 \frac{(x + 0.0049995 \text{ e-}2)(x + 115.07)(x - 115.12)}{(x + 0.0049975)(x^2 + 0.62948 \text{ e-}6 x + 0.040020)} \end{aligned}$$

このままでは不必要な極を生じるが、近似 GCD を取り除くと、

$$\tilde{r}(x) \simeq \frac{1.0}{24.998x^2 + 1.0004}$$

となり元の関数に近い有理関数が得られることが分かる。

同じことが、 $r_{4,4}(x), r_{5,5}(x), \dots$ にも見られる。 $r_{4,4}(x)$ では共通因子は 2 次の多項式となり、 $r_{5,5}(x)$ では共通因子は 3 次の多項式となる。そして、ハイブリッド有理関数近似を行うことで分子分母が 2 次の元の有理関数に一致するような有理関数に変換される。

4.4.2 $\log(x+2)$ の場合

このような入力 of 誤差や Gauss 消去法の計算過程で生じる誤差は、補間区間内で対称関数でない \log のような関数を近似した場合にも影響してくると思われる。しかしこのような関数では、有理関数の分子分母の次数が何次であっても、厳密計算を行った場合にランク落ちが生じない。しかし、同じ有理関数による近似でも、有効桁数によって Gauss 消去法の計算過程には変化がみられる。例として、 $\log(x+2)$ を補間区間 $[-1, 1]$ において $r_{3,3}(x)$ で近似することを考える。

1. 有効桁数が小さい場合

有効桁 3 桁で計算を行ったとする。条件数の部分でも述べたように、条件数は残差が何倍拡大されて相対誤差に反映するかの倍率となっている。したがって、得られた近似解にある程度の精度を持たせるためには、条件数に対して十分な大きさの有効桁を用意して計算を行う必要がある。ここで、有効桁 3 桁というのは、 $r_{3,3}(x)$ での有理関数近似の問題の条件数が 10^7 であることを考えれば、明らかに

精度が不足しているといえる。このように、条件数に対して不十分な精度で計算を行うと、次のような結果が得られる。

有効桁3桁で Gauss 消去法を適用した場合、三角化された行列は次のようになる。

$$\left[\begin{array}{ccccccc|c} 1.0 & -1.0 & 1.00 & -1.00 & 0 & 0 & 0 & 0 \\ 0 & 2.00 & 0 & 2.00 & -1.10 & -1.10 & -1.10 & 1.10 \\ 0 & 0 & -1.00 & 0.00100 & 0.550 & 0.549 & 0.549 & 0.144 \\ 0 & 0 & 0 & -0.371 & -0.0421 & 0.176 & 0.321 & -0.00158 \\ 0 & 0 & 0 & 0 & 0.0259 & -0.0729 & 0.283 & 0.00639 \\ 0 & 0 & 0 & 0 & 0 & 0.00362 & -0.0109 & 0.00279 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.00633 & -0.000298 \end{array} \right]$$

最後の行をみると、有効桁3桁に対して微小な値を持つ成分 -0.000298 が存在する。ここで、後退代入によって得られる有理関数は、次のようになる。

$$\begin{aligned} r_{3,3}(x) &\simeq \frac{0.693 + 2.25x + 1.57x^2 + 0.0133x^3}{1 + 2.53x + 0.629x^2 - 0.0471x^3} \\ &= 0.282 \frac{(x + 117.)(x + 1.00)(x + 0.447)}{(x + 2.86)(x - 16.7)(x + 0.446)} \end{aligned}$$

最後の項 $(x + 0.447)$ と $(x + 0.446)$ が近似的な共通因子になっており、これによって得られた有理関数 $r_{3,3}(x)$ は不必要な極を生じる。そこで、近似 GCD を用いてこの共通因子を除去する必要がある。ここで、先ほど示した三角化された行列の最後の行の微小な値となっている成分 -0.000298 を記号 a で置き換えて同様に後退代入を行うと次のような関数が得られる。

$$r_{3,3}(x) \simeq \frac{0.695 + (2.17 - 279.a)x + (1.61 + 128.a)x^2 + (0.135 + 407.a)x^3}{1 + (2.42 - 400.a)x + (0.771 + 475.a)x^2 + 158.ax^3}$$

このままでは、近似的な共通因子は存在しない。ところが、この a に適当な値、例えば $a = 10$ を代入すると、次のような共通因子を生じる。

$$\begin{aligned} r_{3,3}(x) &\simeq \frac{0.695 - 2790.x + 1280.x^2 + 4070.x^3}{1 - 4000.x + 4750.x^2 + 1580.x^3} \\ &= 2.58 \frac{(x + 1.00)(x - 0.000249)(x - 0.685)}{(x + 3.69)(x - 0.000250)(x - 0.685)} \end{aligned}$$

このままでは不必要な極を生じるが、ハイブリッド有理関数近似を行うと分子分母が1次の有理関数になり、元の関数を近似することができる。

2. 有効桁数を大きくした場合

次に、条件数の大きさに対して十分な精度の有効桁を用意して上記のような計算を行う。 $r_{3,3}(x)$ の場合の条件数は 10^7 なので、有効桁15桁で計算を行うことにする。すると、Gauss 消去法で三角化された行列において、先ほど微小な値となっ

ていた成分は、 $-0.166227168749054 \times 10^{-5}$ という有効桁 15 桁に対して十分な大きさを持った値となる。このとき、後退代入によって得られる有理関数は、分子分母に近似的な共通因子を持たない。ここで、この成分を a という記号に置き換えて同様の実験を行うと、 a にどのような値を代入しても、分子分母には近似的な共通因子が現れない。

これらのことから分かるように、近似的な共通因子には、

- 本来はランク落ちを生じるような場合
- 条件数の大きさに対して計算精度が足りない場合

に現れることが分かる。そして、Runge の例のような場合には、近似的な共通因子は、元の関数に近づくために余分な多項式の次数を落す役割を果たしているようにみえる。

5 安定化理論を用いた有理関数補間

5.1 安定化理論

安定化理論 [4] は、不安定なアルゴリズムを安定なアルゴリズムに変換する手法として提案されているものである。代数的アルゴリズムは厳密計算での実行を前提としているため、入力に浮動小数近似した値を利用することで、真の解とは全く異なる値を導出してしまう場合がある。このようなアルゴリズムが不安定なアルゴリズムと呼ばれるもので、代表的なものとしては 0 判定の条件分岐を持つようなアルゴリズムがあげられる。

不安定なアルゴリズムの例を図 7 に示す。図 7 のアルゴリズムに対して $X = 1/3$ を

[入力] : X

[出力] : Zero または NonZero

[アルゴリズム] :

$Y = 3X - 1$

if $Y = 0$ then return Zero
 else return NonZero

図 7: 不安定なアルゴリズム

入力すれば、正しく Zero が返されるが、 $X = 0.3333\dots$ のような近似した値を入力すると NonZero が返される。これは、近似の精度をいくら上げてもし正しい出力 Zero を得ることができない。このように不安定なアルゴリズムでは、浮動小数近似した値を用いることによって条件分岐で誤った方向に進み、真の解に近づくことができなくなってしまうようなアルゴリズムである。

安定化理論は、このような不安定なアルゴリズムを入力に浮動小数近似した値を用いても真の解に近づくことができるような安定なアルゴリズムに変換する手法である。安定化の方法は次のようなものである。

1. アルゴリズムの構造は変えない。
2. データの係数を「区間係数」に置き換える。
3. ゼロ判定の条件文で、「区間係数のゼロ書き換え」を行う。
(区間数が0を含むであれば、その区間を0に書き換える)

一般の精度保証付き数値計算に使用される区間演算では、区間数のゼロ書き換えは行われないが、安定化理論では敢えて区間数のゼロ書き換えを行うことで、条件分岐文で正しい方向に進むことができるようになっている。安定化理論では、真の解に近づくまで有効桁数を上げながら再計算を行う必要がある。有効桁数が小さい場合には、書き換える必要がない区間数までゼロ書き換えしてしまうことで正しい解が得られない場合があるが、有効桁数を上げて繰り返し計算を行うことで、必ず真の解に近づくことが理論的に証明されている [4]。

有理関数近似では、Gauss 消去法のアルゴリズムを安定化することを考える。Gauss 消去法では、明示的なゼロ判定の条件分岐は存在しないが、ランク落ちの判定をするためのゼロ判定が存在しており、安定化が有効なアルゴリズムであると考えられる。

5.2 結果

表 4: 安定化理論を用いた有理関数近似 ($\log(x+2)$, 補間区間 $[-1, 1]$)

○... 不必要な極のない有理関数が得られる

×... 計算過程でランク落ちが生じる

有効桁	$r(5, 5)$	$r(10, 10)$	$r(15, 15)$	$r(20, 20)$
19 桁	○	×	×	×
28 桁	○	×	×	×
38 桁	○	不正確	×	×
48 桁	○	○	×	×
57 桁	○	○	不正確	×
67 桁	○	○	○	×
77 桁	○	○	○	不正確
86 桁	○	○	○	不正確
96 桁	○	○	○	○

表 5: 安定化理論を用いた有理関数近似 ($\cos(5x^2)$, 補間区間 $[-1, 1]$)

有効桁	$r(10, 10)$	$r(15, 15)$	$r(20, 20)$	$r(25, 25)$	$r(30, 30)$
19 桁	×	×	×	×	×
28 桁	○	×	×	×	×
38 桁	○	×	不正確	×	×
48 桁	○	×	○	×	×
57 桁	○	×	○	×	×
67 桁	○	×	○	×	不正確
77 桁	○	×	○	×	○
86 桁	○	×	○	×	○
96 桁	○	×	○	×	○

安定化理論を用いた有理関数近似を行った場合の不必要な極のふるまいを表 4、表 5 に示す。安定化した Gauss 消去法を利用して有理関数近似を行った場合、有効桁数が小さいときにはゼロ書き換えの必要がない区間数までゼロ書き換えされてしまい、計算過程でランク落ちが生じる。一方、有効桁数を上げていくと、解が求まるようになる。始めのうちは求まった解は真の解とは離れた値であり、正しく元の関数を近似することができないが、さらに有効桁を上げれば真の解に収束する。結果としては、このときの有効桁が条件数が安定する有効桁数よりも大きくなっているため、求まった解の値は安定しており、関数近似を行うと不必要な極を生じないような有理関数が得られる。また、表 5 のように厳密には行列式の値が 0 になる場合には、安定化手法を用いると必ずランク落ちが生じ、解が一意に求まらない。これらの結果から、安定化手法を利用すれば、厳密計算の性質に合った結果を得ることができるといえる。

6 まとめ

素朴な有理関数近似を行うことを考えた場合、得られた有理関数には不必要な極が生じることがある。そこで、不必要な極の出現に関する問題について詳しく検討し、具体的に計算を通じて以下のようなことが明らかになった。

1. 有理関数近似で現れる不必要な極は、この極に対応する分子の零点を近似 GCD によって取り除くと高精度な有理関数近似を得ることができる。
2. しかし、不必要な極と対応する零点の位置は有効桁数によって変化する。
3. 有理関数の係数を決定するための係数行列は、極めて悪条件である。
4. 悪条件行列を Gauss 消去法で解いているが、上のような不必要な極を生じる部分以外では極めて良好な近似ができる。

このような状況を検討し、問題を3種に分類することができた。

1. 補間区間内で対称な有理関数から得られるデータ列を厳密計算で近似する場合
Runge の例 $1/(25x^2 + 1)$ を $r_{m,n}(x)$ ($m > 1, n > 2$) の有理関数で近似する問題について検討したところ、係数行列には全ての要素が0となる行が生じる。すなわち、ランク落ちを起こす。このようなランク落ちした行列の対角要素に記号を代入して、有理関数の係数を決定すると、得られる有理関数はこの記号を含むような分子分母の共通因子が生じ、これを簡約化すれば正しく元の有理関数になる。
2. 上の問題を浮動小数計算で近似する場合
厳密な計算では全ての要素が0になる行が存在するが、浮動小数計算では誤差が混入するため、微小な値を持つ要素が現れるためランク落ちせず、係数を一意に決定することができる。有理関数の分子分母の係数は、この微小な値の関数になるが、これが不必要な極と零点の出現に大きく関与する。この微小な値は浮動小数計算の誤差であり、その値は一定ではないが、いずれの場合でもこの値は近似GCDで取り除き得る不必要な極と零点に対応しており、結果の近似には影響を及ぼさない。
3. 一般の関数から得られるデータを浮動小数計算で近似する場合
本論では、 $\log(x+2)$ について検討を行った。結果として、2. と非常に近いふるまいをしていることがわかった。不必要な極と零点の出現、またこれらの結果への関与については2. と同様である。しかし、ここで大きな問題は、これら微小な値が単なる浮動小数計算における近似ではなく、有意な値となっている点である。したがって、有効桁を大きく増加させる場合、2. の微小な値は0に近づくか、一定の値として意味を持ち続けることになる。非常に高い精度で計算を行うと、2. の場合は0に近づくが、この場合は微小な値が意味を持ち始め、不必要な極を持たない有理関数となる。

以上のような問題に対して、安定化理論を用いて計算を行った。この場合には、解が得られるまで有効桁を増加させながら再計算を行うことになる。ここで、解が求まり、それが元の関数の近似になっている場合には、必ず不必要な極は生じない近似になっている。そのため、解を得るためには大きな有効桁が必要となるが、不必要な極のない高精度な近似であることを信頼することができる。また、厳密には行列式が0ならば、解が一意に定まらないことから、厳密計算の性質にあった結果を得ることができるといえる。

参考文献

- [1] 甲斐 博: ハイブリッド有理関数近似の誤差評価, 情報処理学会論文誌, Vol.40, No.4, pp.1754-1759, Apr.1999

- [2] M.T.Noda and H.Kai: Hybrid rational function approximation and its accuracy analysis, *Reliable Computing* 6,pp.429–438,2000
- [3] M.T.Noda, E.Miyahiro and H.Kai: Hybrid rational function approximation and its use in the hybrid integration, in “Advances in Computer Methods for Partial Differential Equations VII”,eds.R. Vichnevetsky, D.Knight and G.Richter, IMACS, pp.565–571,1992
- [4] K.Shirayanagi and M.Sweedler: A Theory of Stabilizing Algebraic Algorithms, *Technical Report* 95-28, Cornell University, 1995, pp.1-92.